AD-A067 756 TEXAS UNIV AT AUSTIN CENTER FOR CYBERNETIC STUDIES F/G 12/2

A NEW OPTIMIZATION METHOD FOR LARGE-SCALE FIXED CHARGE TRANSPOR--ETC(U)

MAR 79 R S BARR, F W GLOVER, D D KLINGMAN N00014-75-C-0616

UNCLASSIFIED CCS-350 NL

1 OF 1

AD
A067756

END

DATE
FILMED
6 -79

DDC

LEVEL II

(12)

AD A067756

DDC FILE COPY

CENTER FOR CYBERNETIC STUDIES

The University of Texas
Austin, Texas 78712

79 04 20 071

12 44p

12

14 CCS - 350

6

A NEW OPTIMIZATION METHOD FOR LARGE-SCALE FIXED CHARGE TRANSPORTATION PROBLEMS

by

10

Richard Barr*
Fred Glover**
Darwin Klingman***

11 Mar[redacted]79

* Assistant Professor and Management Science, Southern Methodist University, Dallas, TX 75275.

** Professor of Management Science, University of Colorado, Boulder, CO 80309.

*** Professor of Operations Research and Computer Sciences, The University of Texas at Austin, BEB 608, Austin, TX 78712.

15

This research was supported in part by Project NR047-021, ONR Contracts N00014-75-C-0616, [redacted] N00014-75-C-0569 with the Center for Cybernetic Studies, The University of Texas. Reproduction in whole or in part is permitted for any purpose of the United States Government.

CENTER FOR CYBERNETIC STUDIES

A. Charnes, Director
Business-Economics Building, 203E
The University of Texas
Austin, TX 78712
(512) 471-1821

406 197

ABSTRACT

This paper presents a branch-and-bound algorithm for solving fixed charge transportation problems where not all cells exist. The algorithm exploits the absence of full problem density in several ways, thus yielding a procedure which is especially applicable to solving real-world problems which are normally quite sparse. Additionally, streamlined new procedures for pruning the decision tree and calculating penalties are presented. We present computational experience with both a set of large test problems and a set of dense test problems from the literature. Comparisons with other codes are uniformly favorable to the new method, which runs more than twice as fast as the best alternative.

1. INTRODUCTION

Fixed charge problems arise in many "integer" and "nonlinear" programming applications. Many of these problems are network problems with fixed charges attached to subsets of the arcs. Examples include the well-known network expansion problems, plant location problems, process selection problems, plus a wide variety of related investment and distribution problems [5, 13, 16, 17, 22, 29, 34, 35, 37, 38, 45, 46, 48, 50, 55]. In each of these, the central decision "to invest or not invest," "to build or not build," "to ship or not ship," can be modeled by imposing fixed charges on appropriate arcs of the network.

This paper examines the important subclass that consists of fixed charge transportation problems with uncapacitated arcs. This subclass is important for several reasons. First, uncapacitated fixed charge problems are generally harder to solve than capacitated fixed charge problems. (Bounds that limit computation are more difficult to come by in the absence of restrictive capacities on the arcs.) Second, every capacitated fixed charge network problem can be transformed into an equivalent uncapacitated problem of the transportation type. The resulting transportation problems

are sparse (i.e., contain few admissible arcs) and have fixed charges on only a portion of the existing arcs. Sparsity is an almost universal feature of real-world transportation problems, including those that are not created as a transformation of another problem. Research to date has primarily focused on totally dense transportation problems with fixed charges on all arcs. Unfortunately, these almost never arise in practical applications. The goal in this chapter, therefore, is to report on experimentation with solution methods designed especially for the more realistic types of fixed charge transportation problems--characterized by less than full density of arcs and fixed charges. Because of the generally greater difficulty of solving uncapacitated fixed charge problems, it is anticipated that the findings will provide conservative estimates of the solution times to be expected for capacitated problems as well. In addition, the study addresses the largest fixed charge problems in the literature, involving up to 1,500 fixed charge arcs.

The procedure proposed for solving fixed charge transportation problems is a special purpose branch-and-bound method. This method implicitly treats the fixed charge restrictions as though modeled by introducing corresponding 0-1 variables, as in standard 0-1 LP formulations of fixed charge problems. However, such variables are not incorporated into the problem formulation itself, but are handled indirectly by the rules of the solution method.

A chief component of this approach is an imbedded computer routine based on extensive investigations into efficient ways to

solve pure network problems, as described in previous chapters. Thus, in particular, test results indicate that the imbedded routine is from 200 to 400 times faster for solving uncapacitated transportation problems than some state-of-the-art commercial linear programming codes. This points up a second aspect of the study: to determine whether the use of an exceptionally powerful network code can materially facilitate the solution of fixed charge transportation problems utilizing known branch-and-bound principles. Such a determination, of course, rests firmly on the decision rules applied at various "subproblem" levels, and more particularly on the interactions between these decision rules and the imbedded network solution routine. Computational tradeoffs that appear to favor one rule over another when using one solution algorithm may be significantly altered when using another. Moreover, the precise way that a decision rule interfaces with a particular solution algorithm, due to coding considerations, can be an important determinant of overall efficiency. Care has been taken, therefore, not to bypass standard--and even "prosaic"-- approaches in order to make certain that relevant possibilities are given their due. This is reinforced by a concern with the nondense problem structures that have often been neglected by other studies. As a result, roughly twenty different branching and separation rule combinations have been tested to determine their relative merits.

At the same time, to provide links with prior experimentation, a set of totally dense problems [25, 26] have been solved

and compared the results to those of the most efficient method in the literature [32]. The outcomes are uniformly favorable to this new method, which runs more than twice as fast as the best alternative code on all problems of full density. Since no good methods designed to exploit sparse fixed charge transportation structures have been published, comparisons of this approach to others on such problems cannot presently be made. However, comparative results are reported from alternative choice rules incorporated into the new solution routine.

2. PROBLEM FORMULATION

The uncapacitated fixed charge transportation problem may be written:

$$\text{Minimize} \quad \sum_{i\epsilon M} \sum_{j\epsilon N_i} f_{ij}(x_{ij})$$

subject to:

$$\sum_{i\epsilon M_j} x_{ij} = D_j, \quad j\epsilon N,$$

$$\sum_{j\epsilon N_i} x_{ij} = S_i, \quad i\epsilon M,$$

$$x_{ij} \geq 0, \qquad i\epsilon M,\ j\epsilon N_i$$

where M and N are the index sets for the origin and destination nodes, N_i is the index set for those destinations that can receive flow from origin i, and M_j is the index set for those origins that can send flow to destination j. (Thus, there exists arc (i,j) from

node i to node j, and a corresponding flow variable x_{ij}, provided $i \varepsilon M$ and $j \varepsilon N_i$ or equivalently provided $j \varepsilon N$ and $i \varepsilon M_j$.) The quantities D_j and S_i represent demands and supplies, assumed to satisfy $\sum_{i \varepsilon M} S_i = \sum_{j \varepsilon N} D_j$. The function $f_{ij}(x_{ij})$ represents the "fixed charge cost function" for the arc (i,j), where

$$f_{ij}(x_{ij}) = \begin{cases} c_{ij}x_{ij} + F_{ij} & \text{if } x_{ij} > 0 \text{ or} \\ 0 \text{ if } x_{ij} = 0. \end{cases}$$

The fixed charge constant F_{ij} is ordinarily considered to be positive, but is allowed to be zero to accommodate the situation in which only a subset of the arcs have "true" fixed charges. For this subset, the integer programming formulation replaces $f_{ij}(x_{ij})$ by

$$c_{ij}x_{ij} + F_{ij}y_{ij}$$

and appends the constraint

$$x_{ij} \leq U_{ij}y_{ij}$$

where U_{ij} is an upper bound on x_{ij}, and y_{ij} is a 0-1 variable.

In the LP relaxation of the integer problem (where $0 \leq y_{ij} \leq 1$) the optimum value of y_{ij} is clearly x_{ij}/U_{ij}, and furthermore, the updated row equation for a basic y_{ij} variable is obtained from the row equation for x_{ij} by the expression

$$y_{ij} = x_{ij}/U_{ij} + \delta_{ij}/U_{ij}$$

where δ_{ij} is the slack variable for $x_{ij} \leq U_{ij}y_{ij}$.

3. HISTORY AND SURVEY OF EXISTING METHODOLOGIES

An early paper by Hirsch and Dantzig [27] established several important characteristics of the general fixed charge linear programming problem or, as it has come to be known, "the fixed charge problem." It was shown that the feasible region for this problem is a bounded convex set and that its optimal solution will be at an extreme point of this set. The paper also showed that not only is a local minimum solution not necessarily a global minimum but that every basic feasible solution is a local minimum in non-degenerate problems with all positive fixed charges.

3.1 Heuristic Approaches

The earliest solution methods in this area were heuristics for the fixed charge transportation problem. The first, by Balinski [1], operated under the assumption that typical problems had many fewer origins than destinations and, therefore, each demand will tend to be satisfied by a single origin. For this case, a good linear under approximation to $f_{ij}(x_{ij})$ is $c_{ij}x_{ij} + F_{ij}x_{ij}/(\text{minimum}\ \{S_i, D_j\})$, since the two objective functions' values coincide when the arc's flow is at its upper or lower bound. Balinski's approach used this linear relaxation of the functional with the transportation constraints to obtain an approximate solution to the original problem.

Another heuristic method for the fixed charge transportation problem, by Kuhn and Baumol [36], assumed that all typical fixed charges are approximately the same, and used a modification

of Vogel's approximation method to attempt construction of a good, but degenerate, solution. Other approximation algorithms have been proposed for the fixed charge problem by Cooper and Drebes [9], Denzler [11], and Steinberg [49].

More recently, Walker [54] proposed a procedure for the fixed charge problem which he aptly describes as follows:

> The basic approach in all three variants of the algorithm is (1) to obtain a local minimum by using the simplex method with a modification of the rule for selection of the variable to enter the basic solution, and (2) once at a local optimum to search for a better extreme point by jumping over adjacent extreme points to resume iterating two or three extreme points away. [54]

Good computational results were reported, with the heuristic solutions often being optimal. This algorithm is discussed again in a later section.

3.2 Optimal Solution Methods

The first optimizing algorithm published specifically for fixed charge problems was by Murty [40] in 1967. Beginning with a feasible solution, his procedure determines a progression, or ranking, of problem bases with increasing total variable cost components in their objective functions. Each basis added to the progression is chosen from the set of extreme points which are adjacent to the previously-ranked bases and whose objective function values fall within stated bounds. This ranking terminates when an upper bound on the total variable cost is reached, at which point the ranked vertex with the smallest value in terms of the original fixed

charge objective function is optimal. While the algorithm is finite, even strong bounds on the total variable cost can lead to an extremely large number of bases to search and store, explicitly or implicitly, for each ranking. (Such was the case in McKeown's testing of a computer code based on this approach [39].) Weak bounds simply exacerbate the difficulty.

In the same year, Gray [26] proposed an alternative optimization approach which implicitly enumerates all possible patterns of "off-on" states of the fixed charge variables. The enumeration process was restricted by strengthening bounds on the total fixed charges and by exploiting the underlying transportation structure. This approach has many of the same implementational problems as Murty's, but for the first time an optimizing code for fixed charge transportation problems was devised. Gray also proposed and solved [25] a series of problems which are still in use as benchmarks by researchers.

Soland [47] later developed an efficient algorithm and computer code for the plant location problem with concave costs, of which the fixed charge transportation problem is a special case. Within a branch-and-bound framework, he uses piecewise linear envelopes to underestimate the concave objective function in the problem relaxations which are capacitated transportation problems. The algorithm stops with a solution whose value is guaranteed to be within a stated percent of the optimal, and good computational results were reported on Gray's test problems with a tolerance of one

percent.

Kennington and Unger [32] also used a branch-and-bound approach for solving fixed charge transportation problems and derived an efficient means of calculating the simple penalties. A computer code based on their algorithm was used to solve totally dense problems with dimensions of up to 5 x 20. Their test results on fixed charge transportation problems are the best published to date.

Kennington used this code in [31] to study the effects on problem solvability of various branching and separation rules, penalties, problem dimensions, and cost characteristics of the optimal solution. Also, Rardin and Unger extended this branch-and-bound algorithm to fixed charge network problems and used analysis of variance in their testing to study the effects on solution time of: the size of fixed costs relative to variable costs, solution method, and number of arcs in the problem with fixed charges [41]. (Some of these researchers' findings are reexamined in Section 5.5.)

Other researchers on related fixed charge and concave cost problems have used dynamic programming [56] and implicit enumeration or branch-and-bound [15, 30, 42, 49].

4. NEW ALGORITHM FOR THE FIXED CHARGE TRANSPORTATION PROBLEM

As emphasized previously, the algorithms and computer codes described in the literature have not addressed the more real-

istic large scale and sparse fixed charge problems. This section presents a new optimization algorithm designed explicitly for this problem type and describes FIXNET, an experimental code based on this new approach.

In the discussion that follows, it is assumed that the reader is familiar with the branch-and-bound technique of integer programming. Algorithmic descriptions follow the terminology and structure of Geoffrion and Marsten [18].

4.1 Solution Method

The general solution method is a branch-and-bound procedure utilizing the LIFO ("last in first out") rule for constructing the enumeration tree. The LIFO rule was selected because it is extremely simple and easy to work with. Consequently attention can be focused on other types of decision rules and tests whose relative influence appear likely to be independent of the rules for constructing the enumeration tree. (The ability to produce a superior method using LIFO more clearly establishes the merits of the other components of the algorithm.) In particular, the goal is to demonstrate that a collection of straightforward techniques, properly integrated with a powerful method for solving transportation subproblems, can prove highly effective for solving fixed charge transportation problems with a variety of structures.

In addition to LIFO, the other "structural" ingredients of the algorithm are as follows.

4.2 Linear Relaxation and Current Arc Costs

The standard linear relaxation of the fixed charge problem (see Balinski [1]) is used, thus $f_{ij}(x_{ij})$ is replaced in the objective function by $d_{ij}x_{ij}$ where $d_{ij} = c_{ij} + F_{ij}/U_{ij}$, making use of the previously noted fact that $y_{ij} = x_{ij}/U_{ij}$ holds at optimality in the relaxed integer programming formulation. By means of this connection we further observe that y_{ij} is strictly between its upper and lower bound if and only if the same is true of x_{ij}. Also, by means of the earlier observation that the updated LP row equation for y_{ij} can be obtained from that of x_{ij} (for $x_{ij} > 0$ and hence basic) by the relation

$$y_{ij} = x_{ij}/U_{ij} + \delta_{ij}/U_{ij}$$

we may restrict attention to the flow and basis conditions of x_{ij} to know automatically the corresponding conditions for y_{ij}.

Since we deal with an uncapacitated problem the relaxation that replaces $f_{ij}(x_{ij})$ with $d_{ij}x_{ij}$ is somewhat weaker than might otherwise be the case, for reliance is made on the highly redundant bound U_{ij} for x_{ij} given by $U_{ij} = \min\{S_i, D_j\}$. (Useful tighter bounds appear to entail substantial computational expense, though one possibility is to use the accelerated "dual start" methods of the [23] paper, which are amenable to post-optimization for multiple computations.)

Thus, the "current cost" attached to arc (i,j) at a given stage of the branch and bound process is:

d_{ij} if x_{ij} is <u>free</u>,

M (large positive constant), if x_{ij} is constrained to 0, or

c_{ij}, if x_{ij} is constrained to positivity.

In the last case, the constant F_{ij} is added to the current objective function value (for each arc so constrained) and x_{ij} is permitted to receive any non-negative value so that it may in fact receive the value zero. This does no harm, since allowing x_{ij} to be zero while incurring the fixed charge F_{ij} merely produces an "inferior solution" (which is conspicuously worse than the one in which x_{ij} is zero and the fixed charge is absent). The inferior solution will either be bypassed or identified as dominated during the solution process. However, a special technique for taking advantage of the fact that x_{ij} should be positive will be described subsequently.

4.3 Algorithm for Transportation Subproblems

The method for solving transportation problems is an outgrowth of several years of continuing developmental and testing efforts documented in [2, 3, 19, 20, 23, 24]. The procedure makes use of the "augmented thread" [24] and is about three or four times faster than the PTRANS code reported in [20]. Like PTRANS, it is based on a network specialization of the primal simplex method. Thus no dual steps are made, despite the widespread use of the dual method in branch-and-bound algorithms.

This particular code by Barr [2] was chosen over the out-of-kilter approach for two reasons: speed and memory require-

ments. Not only is it one of the fastest transportation codes in existence, it has the smallest memory requirements of any known code for this problem. Specifically, only four node-length arrays and two arc-length arrays must be stored as opposed, for example, to four and eleven, respectively, for the SUPERK code [4]. These spacial parameters are exceptionally important in this context since additional fixed-charge values and the branch-and-bound tree structure require another five arc-length arrays.

4.4 Fathoming Tests

As is customary, in branch-and-bound algorithms, the following conditions terminate forward progress through the enumeration tree and initiate a "backtracking" step:

(1) The current subproblem has no feasible solution.

(2) The optimal objective function value for the current subproblem (augmented by "penalties" and added to the fixed costs inherited by that subproblem) exceeds or equals the objective function value of the best feasible fixed charge solution currently found. (Note that this condition actually includes (1) in a primal approach, since infeasibility is detected by a "big M" variable which is positive in an optimal solution.)

(3) The optimum objective function value for the current subproblem does not worsen (become larger) when the optimal solution is evaluated by the true fixed charge objective function (instead of the linearized objective function).

In case (3), the current solution can provide a new "best known" solution for the original problem. More generally, any time the current solution provides an improved evaluation according to the true fixed charge objective, then the solution becomes the new "best known" (incumbent) solution, regardless of the relation of this evaluation to that of the linearized objective function.

4.5 Culling of Variables

In addition to these standard fathoming tests, we make use of the following. As established in [27], an optimal fixed charge solution can be found among solutions that are basic. Hence, the variables x_{ij} that are "constrained to positivity" can be further required to be members of the basis. In a network, a collection of variables can be basic only if their arcs do not form a loop. Each time a variable is to be constrained to positivity, therefore, the procedure tests whether any loops are created relative to other variables so constrained. If so, the constraining step is identified as inadmissible. The primary virtue of this test is that it can be executed rapidly even in very large networks with a very small commitment of memory resources.

4.6 Penalty Calculations

The algorithm employs the simple "up" and "down" (one dual pivot "look ahead") penalties as proposed by Driebeek [12]. Although commonly used in integer programming for branching decisions,

strengthening of bounds for fathoming, and identifying variables for "pegging" positive or zero, these values are laborious to calculate by standard means.

The application of penalties to fixed charge network problems was first undertaken by Kennington and Unger [31, 32] who used group theory to derive a specialization of the Charnes' "poly-ω" [6] technique for efficiently calculating individual penalties. This technique computes the pair of penalties for a given basic fixed charge arc by repricing the basis and inspecting all nonbasic variables. Thus, to determine the penalties associated with all basic fixed charge variables involves several passes of all of the nonbasic arcs.

The new algorithm uses a "generalized predecessor" labeling approach that permits all of the penalties to be determined (in the network structure) with only moderate effort beyond that required to calculate <u>one</u> of these penalties. The details of this procedure are given in the Appendix, but the essential outcome is to enable the "poly-ω" step to be implemented for all constraints simultaneously via a set of easily-determined node labels. These labels allow ready identification of all fixed charge arcs in the "stepping stone" path of each nonbasic arc. Thus, with this information, only one pass of the arc data is required to perform the minimum ratio calculations for all fixed charge penalties, as opposed to multiple passes using other techniques.

4.7 "Local Star" Optimum Advanced Start

The computational effort of the solution process diminishes as tighter bounds are placed on the optimum solution value. Therefore, a smaller upper bound on the optimum was sought by the initial use of a pivoting procedure proposed by Walker [54] for isolating a "local star," or locally optimum basic solution (also see [6]).

This modified simplex pivoting procedure prices out a given nonbasic arc by using its basis representation to determine the change in the nonlinear objective function value that would be made by pivoting the variable into solution. Specifically, the nonbasic's "stepping stone" path must be traced to calculate the leaving arc, the change in total variable cost, and the changes in total fixed cost from degenerate basic arcs becoming positive and from ties for the leaving variable. With this information any change in the fixed charge objective function is known and the pivot may be performed if it is improving.

Although every extreme point in the fixed charge problem is a local optimum in the absence of degeneracy and negative fixed charges [27], network problems are notoriously degenerate and good results were reported by Walker with this technique. Hence, this optional starting procedure was included in the experimental code for testing.

5. COMPUTATIONAL TESTING OF THE NEW ALGORITHM

5.1 Computer Code Description

All of the algorithmic features described above were implemented in a new optimization code called FIXNET. FIXNET is written in a machine-independent FORTRAN IV and was given a modular design to facilitate experimentation with various decision rules and parameters. It maintains all data in internal storage and uses no special coding to capitalize on the features of a particular machine.

This section describes the initial testing of FIXNET to determine the best operating rules plus additional testing on a variety of problem types to measure the effect of problem parameters on solvability. Included in the test set are sparse problems and the largest fixed charge problems to appear in the literature, with the equivalent of up to 1,500 integer variables.

All problems were run on a CDC 6600 computer with FIXNET compiled by the RUN compiler. All times given are exclusive of input and output and indicate the total central processing time required to optimize a particular problem.

5.2 Advanced Starting Procedure

In the initial testing stage, the use of the "local star" pivoting scheme was evaluated and found to not be cost effective. In most cases, Walker's "phase 1" procedure did not substantially improve the solution derived from the Balinski relaxation and the pricing procedure for nonbasics was computationally expensive, even though it was specialized to transportation problems and used the

efficient ATI data structure. This may have been a consequence of low problem degeneracy and the tendency towards local optima at extreme points described previously. As a result, this procedure for obtaining a stronger initial bound was not used in the final version of FIXNET, which determines an initial solution from Balinski's relaxation.

5.3 Branching and Separation Strategies

Within the branch-and-bound framework, two rules for directing the search among subproblems are crucial to the effective performance of an algorithm. First, following the solution of a subproblem that cannot be fathomed, a separation variable is chosen, usually from the set of basic integer variables which are between their upper and lower bounds. Two new "candidate" subproblems are derived by placing contradictory constraints on the separation variable, such as requiring it to be zero and to be positive.

Secondly, the branching decision rule determines which one of the outstanding candidate subproblems is to be solved next. Within the LIFO strategy used here, the next subproblem is one of the two new candidates formed by separation or, after fathoming, the most recently defined candidate problem that is still eligible.

In total, approximately twenty separation and branching rule combinations were tested. The separation variables were always chosen from the basis and separation accomplished by means of the cost parameterization procedure described in Section 4.2.

Of the two general strategies tested, the first utilized the up and down penalties for all basic fixed charge arcs between their upper and lower bounds. Separation criteria tested included choosing the variable with the largest individual penalty (rule SR-1) and the one with largest absolute difference in penalties (SR-3). A third separation rule considers the smaller penalty for each variable and chooses that variable corresponding to the largest of those penalties (SR-3). Branching rules tested included always forcing the separation variable to its upper bound (BR-1), to its lower bound (BR-2), and in the direction of its smaller or larger penalty (BR-3 and BR-4).

The second strategy set tested did not require the computation of penalties but instead used the function $g_{ij} = F_{ij}$ $(1-(x_{ij}/U_{ij}))$, the deviation of the linear relaxation, d_{ij}, from the true objective function, f_{ij}, at this level of flow. The basic premise is that the deviation values' simplicity and speed of calculation would outweigh any benefits derived from the richer, but computationally more complex, penalty values. The variable chosen for separation had either the largest or smallest deviation (SR-5 and SR-6) and preference could be given to those variables that were not within αU_{ij} units of either bound (SR-7) where $0 < \alpha < 1$. Branching was performed by rules BR-1 and BR-2 above.

While many different test problems were run in order to choose the best overall separation and branching strategy, results are shown here for the small, totally dense test problems posed by

Gray [25]. (Problem specifications are given in Table 1.) The results of running the four best strategies on these problems are shown in Table 2. Note that the use of penalties (rule SR-3) substantially reduces the number of transportation subproblems to be solved and that solution times under this separation rule completely dominate the times obtained with the maximum deviation rule (SR-5).

Table 1

PROBLEM SPECIFICATIONS FOR GRAY'S FIXED CHARGE TRANSPORTATION TEST PROBLEMS

PROBLEM NUMBER	DIMENSIONS	ARCS	TOTAL SUPPLY AND DEMAND	COST RANGES VARIABLE	FIXED
1	3 x 4	12	70	0.64-7.60	5-15
2	4 x 6	24	115	0.59-2.83	10-20
3	4 x 6	24	183	2-114	0-43
4	4 x 8	32	135	0.59-6.99	10-20
5	5 x 7	35	125	0.59-2.83	10-20
6	5 x 7	35	125	0.59-2.83	1-99
7	5 x 7	35	125	0.59-9.01	91-220
8	5 x 7	35	218	2-114	1-36
9	6 x 8	48	160	0.59-6.99	10-20

TABLE 2

FIXNET SOLUTION STATISTICS WITH
SPECIFIED SEPARATION AND BRANCHING RULES

SEPARATION: BRANCHING:	SR-5 BR-1		SR-5 BR-2		SR-3 BR-2		SR-3 BR-3	
PROBLEM	SP	TM	SP	TM	SP	TM	SP	TM
1	7	.016	7	.009	3	.008	3	.016
2	153	.468	140	.426	55	.245	67	.300
3	13	.043	13	.039	1	.008	1	.008
4	67	.260	177	.720	25	.152	32	.188
5	413	1.469	518	1.900	108	.645	95	.555
6	71	.257	33	.124	20	.120	15	.101
7	385	1.450	371	1.489	120	.768	76	.475
8	15	.064	15	.064	1	.016	1	.008
9	1307	6.061	722	3.297	357	2.814	305	2.392

SP = number of transportation subproblems solved

TM = CDC 6600 central processor seconds required for optimization

The use of the streamlined penalty calculation scheme was deemed cost-effective for FIXNET and strategy SR-3/BR-3 was adopted as the standard for all test runs made for this study.

5.4 Comparison of Solution Codes

The test problems posed by Gray in 1968 have been solved

by numerous researchers in this area, thus providing a limited means of comparing a variety of solution approaches. Table 3 provides comparative solution times on the Gray problems for the following codes: Gray's decomposition code [26], Fisk and McGowan [14], McGowan's extreme-point ranking approach [39], the branch-and-bound codes by Soland [47] and Kennington and Unger [32], and FIXNET. The testing environments differ, and the Univac 1108 and CDC 6600 computers have comparable add times, while the Univac 1110 is twice as fast, and the Burroughs 5500 is three times as slow.

Even with adjustments for different machine efficiencies, FIXNET clearly dominates all other codes on all problems. Although FIXNET was designed for larger sparse problems, it outperformed the reported solution times of the two best codes in the literature (Kennington's and Soland's) by factors of 1.33 to 6.63 with an average of 3.7 times faster than Kennington's on these problems.

Although Kennington and Unger's code uses the same initial problem relaxation as FIXNET, the number of subproblems differed, at times greatly. This difference is the result of different branching, separation, and candidate problem selection rules. It is particularly interesting to note that in seven out of the nine cases shown here, the "elementary" LIFO technique required fewer subproblems for optimization than the sophisticated "best candidate" selection technique of Kennington and Unger. Moreover, even when FIXNET examined a larger number of subproblems, its solution times were favorable due to the speed in which the subproblems can be

TABLE 3

COMPARATIVE CODE STATISTICS ON GRAY'S FIXED CHARGE TRANSPORTATION PROBLEMS

CODE:	GRAY	FISK AND MCKEOWN	MCKEOWN[2]	SOLAND[4]	KENNINGTON AND UNGER		FIXNET	
MACHINE:	B-5500	UNIVAC 1110	UNIVAC 1108	CDC 6600	CDC 6600		CDC 6600	
PROBLEM	TM	TM	TM	TM	SP	TM	SP	TM
1	7.7	.4	.54	DNR	6	.028	3	.016
2	32.6	7.3[1]	DNR[3]	0.4	96	.746	67	.300
3	26.3	3.5	1.31	DNR	4	.030	1	.008
4	171.4	84.5	DNR	DNR	48	.601	32	.188
5	263.8	195.1	DNR	1.7	224	2.760	95	.555
6	146.9	33.9	DNR	DNR	46	.538	15	.101
7	97.0	42.9	DNR	DNR	56	.836	76	.475
8	3,262.8	DNR	2.22	DNR	4	.053	1	.008
9	1,510.0	200.+	DNR	7.0	62	8.890	305	2.392

TM = solution time in CP seconds

SP = number of subproblems solved

DNR = did not report

[1] Suboptimal solution

[2] Fixed charge linear programming code

[3] Data storage overflow

[4] Solution value within 1 percent of optimum, using separable concave transportation code

solved and the efficiency in which the decision rules could be implemented. This preliminary testing indicates not only the efficiency of the code, but the narrow variation in solution time points out its robustness.

5.5 Computational Studies of Problem Parameters

While many algorithms for the fixed charge transportation problem have appeared in the literature, computational testing has been restricted to extremely small problems. In fact, the largest such problem reported optimized to date has dimensions of 5 x 20 with 100 fixed charge arcs [31].

In order to test the efficiency of FIXNET on larger and sparser problems and to explore the impact of topological characteristics on tractability, a large number of new test problems were devised. This was accomplished using a modified version of NETGEN, a machine-independent program for generating randomized network problems [33]. (Copies of this program are available to permit duplication of test problems by other researchers.)

Because of the peculiar nonlinear topology of integer and fixed charge programs, widely varying degrees of difficulty may be encountered in optimizing seemingly similar problems. (Note the solution times for Gray's 5 x 7 problems in Table 3.) Researchers in this area have put forward numerous explanations for problem tractability, based on both a priori analysis of problem characteristics and a posteriori evaluation of optimum solutions. Presented

in this section are three studies that use FIXNET-generated statistics to test proposed explanations for what makes a problem difficult to solve.

Problem set A was designed to evaluate the effect on solution times of differences in average variable and average fixed costs. Within the set were two groups of problems which were identical except for the fixed charge values. As shown in Table 4, all problems were 50 x 150 with 1500 arcs, a total supply of 2500, and unit costs with values from 0 to 10. The first four problems were generated with fixed charges on 300 arcs and the second four had 600 fixed charge arcs. Each problem within a group of four had different ranges for the fixed cost, with the narrowest range of 0 to 50 and the widest range of 0 to 10,000.

The effect of average problem fixed cost on FIXNET solution time was negligible. As indicated in Table 4, times ranged from 4.150 to 4.966 seconds on the first group and 13.901 to 15.551 seconds on the second group, with no discernable relationship between average fixed cost and solution time. However, the number of fixed charge variables had a significant effect on solvability.

In the computational study by Kennington [31], _a posteriori_ analysis of optimal solutions suggested another possible determinant of solution difficulty. As the ratio of the total fixed charges to the total variable cost in the optimal solution decreased, solution times tended to rise. Hence, problems became much more difficult when the variable costs dominated. To verify

this observation, a basic test problem was constructed and the total supply modulated so as to vary the amount of flow in the problem and, therefore, the total variable cost and its proportion to total fixed cost. As indicated in Table 5, adjustment of the total supply varied the ratio from 7,257 to .36; solution times fluctuated, but without the relationship described above. Similar test sets yield the same inconclusive results. The inability to verify Kennington's observation may be the result of his use of small, totally dense problems plus a different branch-and-bound algorithm and code.

TABLE 4

PROBLEM SET A:

PROBLEM SPECIFICATIONS AND FIXNET SOLUTION TIMES

PROBLEM SIZES: 50 x 150 20% DENSE (1500 ARCS) TOTAL SUPPLY: 2500

FIXED CHARGE DENSITY	MAX UNIT: MAX FIXED COSTS	FIXNET SOLUTION TIMES (SEC) ON CDC 6600
20% (300 arc)	10:50	4.966
	10:100	4.405
	10:1000	4.150
	10:10,000	4.480
40% (600 arc)	10:50	15.551
	10:100	13.901
	10:1000	14.457
	10:10,000	14.907

TABLE 5

PROBLEM SET B:

PROBLEM SPECIFICATIONS AND FIXNET SOLUTION STATISTICS

BASIC TEST PROBLEM: 50 x 150, 1500 ARCS
FIXED CHARGE DENSITY: 40% (605 ARCS)
UNIT COSTS 1-10, FIXED COSTS 1-100

PROBLEM	TOTAL SUPPLY	F/C RATIO	SP	PIVOTS	TM
1	1K	7,257	8	1220	5.99
2	2.5K	8,252	12	1616	8.06
3	5K	2,158	77	3163	19.69
4	10K	1,076	114	3788	26.09
5	50K	214.2	49	2364	14.62
6	100K	170.0	146	2338	14.77
7	1M	11.47	130	3714	31.12
8	5M	2.40	189	3725	30.17
9	10M	3.34	2	4490	41.64
10	50M	1.74	2	1199	8.80
11	100M	.36		1155	6.21

F/C Ratio = Ratio of total fixed cost to total variable cost in optimal solution value

SP = number of transportation subproblems solved by FIXNET

Pivots = number of pivots made in solving subproblems

TM = FIXNET solution time in CDC 6600 central processor seconds

Note, however, that the solution times presented here are for mixed integer programming problems with 1500 continuous variables and 600 fixed charge variables. Even the largest time of 31 seconds is remarkably good for problems of this size and the range of solution times emphasizes the efficiency and robustness of the new algorithm.

Test Set C, described in Table 6, was designed to study the effect of problem dimensions holding the number of fixed charge variables constant. With these problems, the average solution times increased with increasing dimension sizes, as would be expected, but again the average times slightly more than doubled when the number of fixed charge variables was doubled. Since solution time for combinatorial problems often increases geometrically with the number of integer or fixed charge variables, the efficiency of this new algorithm is again demonstrated.

In summary, testing has demonstrated that the code FIXNET, based on the new solution algorithm, is extremely efficient for solving fixed charge transportation problems. It has solved problems with 3,000 constraints and 1,200 fixed charge variables in nine seconds on a CDC 6600, and is designed for the sparse large-scale problems encountered in real-world applications. This procedure brings many heretofore unsolvable problems within the reach of practitioners and researchers alike, and in general provides a tool for handling the realistic considerations of fixed charges and economies of scale that have previously been neglected in network

TABLE 6

PROBLEM SET C:

PROBLEM SPECIFICATIONS AND FIXNET SOLUTION STATISTICS

DIMENSIONS	NO. ARCS	NO. FIXED CHARGE ARCS	SP	TIME
30 x 70	300	300	2	.607
	600		62	6.693
	900		8	2.236
	1200		2	1.973
50 x 150	750		2	3.151
	1500		2	5.150
	3000		2	6.513
30 x 70	600	600	50	7.424
	1200		8	3.253
	2000		2	3.724
50 x 150	600		2	2.287
	1200		192	28.841
	2000		8	8.453
	3000		3	11.987
100 x 300	3000		2	14.869
50 x 150	3000	1200	4	9.029

RANGE OF UNIT COSTS: 1 - 10 FIXED COSTS: 1 - 10

TOTAL SUPPLY: 100 TIMES THE NUMBER OF SOURCES

SP = number of transportation subproblems solved

TIME = solution time in CP seconds on CDC 6600

applications due to the computational difficulty ordinarily viewed as inherent in such considerations.

APPENDIX

THE GENERALIZED PREDECESSOR LABELING METHOD FOR COMPUTING PENALTIES IN FIXED CHARGE TRANSPORTATION PROBLEMS

The principal requirement of Driebeek-type penalty calculations is to know the updated simplex row coefficients for each row associated with a basic integer variable. To obtain such coefficients for all rows of interest by a <u>single pass</u>, we use the following <u>generalized predecessor</u> labeling procedure. Define the path $P(q,s)$ to be the ordered set of nodes and links connecting and including nodes q and s in the rooted spanning tree basis. Hence, $P(q,s) \equiv q(q,h),h,\ldots,k,(k,s),s$. Therefore, nonbasic arc (q,s) is represented in the basis by the arcs corresponding to the links in $P(q,s)$, with the arcs' directions determined from their origin-to-destination orientation.

The coefficient of x_{qs} in the updated row equation for a basic variable x_{ij} is therefore zero if arc (i,j) does not lie on $P(q,s)$. If the basic arc (i,j) does lie on $P(q,s)$, the row coefficient of x_{qs} is +1 if the basic arc is oriented in the same direction as its corresponding link in $P(q,s)$ and is -1 if the basic arc is oriented in the opposite direction. By the connection between y_{ij} and x_{ij}, the coefficient of x_{qs} in the row equation of the integer fixed charge variable y_{ij} is therefore, correspondingly, 0, $1/U_{ij}$ or $-1/U_{ij}$.

Therefore, by tracing out the stepping stone path (basis equivalent path) for each nonbasic variable x_{qs}, all nonzero tableau row coefficients for this variable in the row equation for each basic x_{ij} (and hence y_{ij}) can be determined. This path is found for a given nonbasic arc (q,s) by tracing the predecessors of q and s to their point of intersection.

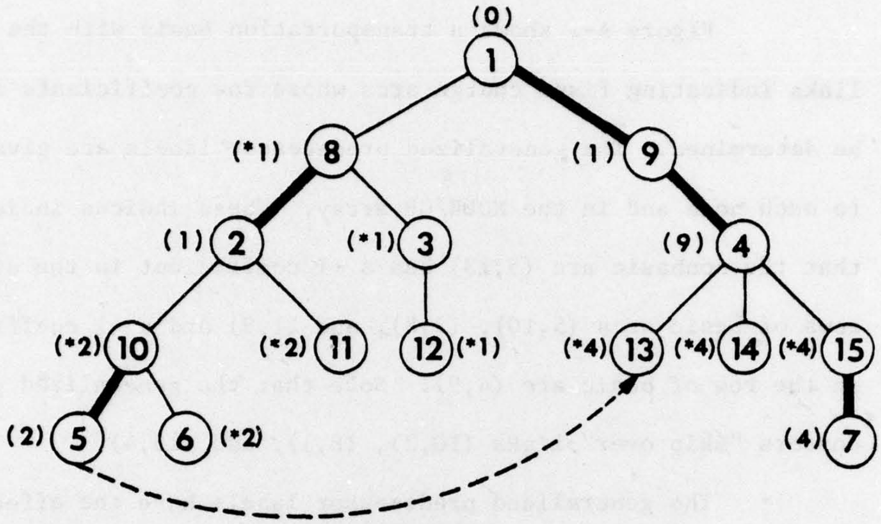
The generalized predecessor labeling method takes advantage of the preceding observations as follows.

Let p(q) be the predecessor of node q or zero if q = r (the root node of the basis tree), and let I be the set of basic fixed charge arcs with flow between their upper and lower bounds. The <u>generalized predecessor</u> label for node q is defined as: (a) zero if q = r, (b) the first node $q^* \neq q$ in P(q,r) such that $(q^*,p(q^*)) \varepsilon I$ or $(p(q^*),q^*) \varepsilon I$, or (c) r, if q^* does not exist. (In case (b), node $h \varepsilon P(q,r)$ "preceeds" node $k \varepsilon P(q,r)$ if $d(h) > d(k)$, where d(h) is the number of nodes in P(h,r).) This value is flagged with a "*" if $(q,p(q)) \notin I$ and $(p(q),q) \notin I$.

The fixed charge arcs in the basic representation of nonbasic arc (q,s) are recovered by tracing the generalized predecessors of nodes q and s to their point of intersection. For each node k encountered in this tracing, the link (k,p(k)) corresponds to a basic fixed charge arc, except when node k is flagged with "*" or is the root node. As before, the orientation of the basis arc with respect to its link determines the value of the basis row coefficient.

Figure A-1 shows a transportation basis with the bolder links indicating fixed charge arcs whose row coefficients are to be determined. The generalized predecessor labels are given next to each node and in the NODE/GP array. These indices indicate that the nonbasic arc (5,13) has a +1 coefficient in the simplex rows of basic arcs (5,10), (2,8), and (1,9) and a -1 coefficient in the row of basic arc (4,9). Note that the generalized predecessors "skip over" links (10,2), (8,1), and (13,4).

The generalized predecessor labels have the effect of logically compressing the basis tree to include only those fixed charge arcs of interest. By so doing, the work required to compute the full set of penalties is greatly reduced over other techniques.



NODE	GP
1	0
2	1
3	*1
4	9
5	2
6	*2
7	4
8	*1
9	1
10	*2
11	*2
12	*1
13	*4
14	*4
15	*4

Bolder links indicate fixed charge arcs with flow between upper and lower bounds.

Figure A-1 - Optimal Rooted Spanning Tree Basis for 7x8 Transportation Problem with Generalized Predecessor Node Labels

REFERENCES

1. Balinski, M. L. , "Fixed Cost Transportation Problems," Naval Research Logistics Quarterly 8 (1961), 41-54.

2. Barr, R. S., "Streamlining Primal Simplex Transportation Codes," Research Report to appear Southern Methodist University, Cox School of Business, Dallas, Texas 75275.

3. Barr, R. S., F. Glover, and D. Klingman, "An Improved Version of the Out-of-Kilter Method and a Comparative Study of Computer Codes," Mathematical Programming 7, 1 (1974), 60-87.

4. Barr, R. S., F. Glover, and D. Klingman, "Enhancements to Spanning Tree Labeling Procedures for Network Optimization," Research Report CCS 262, Center for Cybernetic Studies, The University of Texas at Austin, Austin, Texas 78712 (1976).

5. Bod, P., "The Solution of a Fixed Charge Linear Programming Problem," in Harold W. Kuhn, ed., Proceedings of Princeton Symposium on Mathematical Programming, Princeton, N.J.: Princeton University Press, 1970, pp. 367-375.

6. Charnes, A. and W. W. Cooper, Management Models and Industrial Applications and Linear Programming, Volumes I and II, New York: Wiley, 1961.

7. Charnes, A. and W. W. Cooper, "The Stepping Stone Method of Explaining Linear Programming Calculations in Transportation Problems," Management Science 1,1 (1954), 49-69.

8. Charnes, A., and M. Kirby, "The Dual Method and the Method of Balas and Ivanescu for the Transportation Model," Cahiers du Centre d'Etudes de Recherche Operationelle 6,1 (1964).

9. Cooper, L. and C. Drebes, "An Approximate Solution Method for The Fixed Charge Problem," Naval Research Logistics Quarterly 14,1 (1967), 101-113.

10. Dakin, R. J., "A Tree Search Algorithm for Mixed Integer Programming Problems," Computer Journal 8,3 (1965), 250-255.

11. Denzler, David R., "An Approximative Algorithm for the Fixed Charge Problem," Naval Research Logistics Quarterly 16,3 (1969), 411-416.

12. Driebeek, N.J., "An Algoithm for the Solution of Mixed Integer Programming Problems," Management Science 12,7 (1966), 576-587.

13. Dutton, Ron, George Hinman, and C. B. Millham, "The Optimal Location of Nuclear Power Facilities in the Pacific Northwest," Operations Research 22,3 (1974), 478-487.

14. Fisk, John C. and Patrick G. McKeown, "A Storage Efficient Algorithm for the Fixed Charge Hitchcock Transportation Problem," Research Report, State University of New York at Albany, Albany, NY.

15. Florian, M. and P. Robillard, "An Implicit Enumeration Algorithm for the Concave Cost Network Flow Problem," Management Science 18,3 (1971), 184-193.

16. Frank, Howard and Ivan T. Frisch, Communication, Transmission and Transportation Networks, Reading, Mass.: Addison-Wesley Pub. Co., 1971.

17. Garver, Leonard L., Economic Scheduling of Electric Power Generators by Integer Programming, Ph.D. dissertation, Northwestern University, 1961.

18. Geoffrion, A.M. and R.E. Marsten, "Integer Programming Algorithms: A Framework and State-of-the-Art Survey," in A.M Geoffrion, ed., Perspectives on Optimization: A Collection of Expository Articles, Reading, Mass.: Addison-Wesley, 1972.

19. Glover, R., D. Karney, and D. Klingman, "The Augmented Predecessor Index Method for Locating Stepping-Stone Paths and Assigning Dual Prices in Distribution Problems," Transportation Science 6,1 (1972), 171-181.

20. Glover, F., D. Karney, D. Klingman, and A. Napier, "A Computational Study on Start Procedures, Basis Change Criteria and Solution Algorithms for Transportation Problems," Management Science 20,6 (1974), 793-819.

21. Glover, F. and D. Klingman, "Double-Pricing Dual and Feasible Start Algorithms for the Capacitated Transportation (Distribution) Problem," University of Texas at Austin, Austin, TX (1970).

22. Glover, F. and D. Klingman, "Network Applications in Industry and Government," AIIE Transactions 9,4 (1977), 363-376.

23. Glover, F., D. Klingman, and A. Napier, "An Efficient Dual Approach to Network Problems," OPSEARCH 9,1 (1972), 1-19.

24. Glover, F., D. Klingman, and J. Stutz, "The Augmented Threaded Index Method for Network Optimization," INFOR 12,3 (1974),

25. Gray, Paul, "Mixed Integer Programming Algorithms for Site Selection and Other Fixed Charge Problems Having Capacity Constraints," Technical Report No. 101, Department of Operations Research and Statistics, Stanford University (1970).

26. Gray, Paul, "Exact Solution of the Fixed-Charge Transportation Problem," Operations Research 19 (1971), 1529-1538.

27. Hirsch, W. M. and G. B. Dantzig, "The Fixed Charge Problem," Naval Research Logistics Quarterly 15 (1968), 413-424 (reprinted from 1954).

28. Jambekar, Anil B., "Some Experiments with the Fixed Charge Problem," Research Report, School of Business and Engineering Administration, Michigan Technological University.

29. Jarvis, John J., V. Ed Unger, Ron L. Rardin, Richard W. Moore, "Optimal Design of Regional Wastewater Systems: A Fixed Charge Network Flow Model," Operations Research 26,4 (1978), 538-550.

30. Jones, Arnold P. and Richard M. Soland, "A Branch-and-Bound Algorithm for Multi-Level Fixed Charge Problems," Management Science 16,1 (1969), 67-76.

31. Kennington, Jeff, "The Fixed-Charge Transportation Problem: A Computational Study with a Branch-and-Bound Code," AIIE Transactions 8,2 (1976).

32. Kennington, Jeff and Ed Unger, "A New Branch-and-Bound Algorithm for the Fixed Charge Transportation Problem," Management Science 22,10 (1976), 1116-1126.

33. Klingman, D., A. Napier, and J. Stutz, "NETGEN--A Program for Generating Large Scale (Un)Capacitated Assignment, Transportation, and Minimum Cost Flow Network Problems," Management Science 20,5 (1974), 814-822.

34. Klingman, D., P.H. Randolph, S.W. Fuller, "A Cotton Ginning Problem," Operations Research 24,4 (1976), 700-717.

35. Kraus, Alan, Christian Janssen, Alan McAdams, "The Lock-Box Location Problem," Journal of Bank Research (Autumn 1970), 50-58.

36. Kuhn, Harold W. and William J. Baumol, "An Approximate Algorithm for the Fixed-Charges Transportation Problem," Naval Research Logistics Quarterly 9 (1962), 1-15.

37. Lea, A.C., Location-Allocation Models: A Review, Master's Thesis, University of Toronto, 1973.

38. Levy, Fredinand K., "An Application of Heuristic Problem Solving to Accounts Receiveable Management," Management Science 12,6 (1966), B236-B244.

39. McKeown, Patrick, "A Vertex Ranking Procedure for Solving the Linear Fixed-Charge Problem," Operations Research 23,6 (1975), 1182-1191.

40. Murty, Katta G., "Solving the Fixed Charge Problem by Ranking the Extreme Points," Operations Research 16 (1968), 268-279.

41. Rardin, R. L. and V. E. Unger, "Solving Fixed Charge Network Problems with Group Theory-Based Penalties," Naval Research Logistics Quarterly 23 (1976), 67-84.

42. Reigh, P. and L. G. Barton, "A Non-Convex Transportation Algorithm," in M. L. Beale, ed., Applications of Mathematical Programming Techniques, New York: American Elsevier, 1970.

43. Robers, Philip and Leon Cooper, "A Study of the Fixed Charge Transportation Problem," Report No. COO-1493-9, Washington University, School of Engineering and Applied Science, 1969.

44. Sa, Graciano, "Branch-and-Bound and Approximate Solutions to the Capacitated Plant Location Problem," Operations Research 17 (1969), 1005-1016.

45. Scamell, Richard W., The Application of the Shared Fixed-Charge Assignment Problem to Resource Allocation Problems, Ph.D. Dissertation, University of Texas at Austin, Austin, Texas, 1972.

46. Schoeman, Milton E. F. and Richard W. Scamell, "Control of Interrelated Research and Development Projects," OMEGA 1,6 (1973), 669-678.

47. Soland, Richard, "Optimal Facility Location with Concave Costs," Operations Research 22,2 (1974), 373-382.

48. Spielberg, Kurt, "On Solving Plant Location Problems," in E.M.L. Beale, ed., Applications of Mathematical Programming Techniques, New York: American Elsevier, 1970.

49. Steinberg, David I., "The Fixed Charge Problem," Naval Research Logistics Quarterly 17,2 (1970), 217-235.

50. Stroup, John W., "Allocation of Launch Vehicles to Space Missions: A Fixed-Cost Transportation Problem," Operations Research 15,6 (1967), 1157-1163.

51. Tiplitz, Charles I., "Convergence of the Bounded Fixed Charge Programming Problem, " Naval Research Logistics Quarterly 20,2 (1973), 367-375.

52. Tomlin, J.A., "An Improved Branch and Bound Method for Integer Programming," Operations Research 19,4 (1971), 1070-1075.

53. Tompkins, Curtis J. and V. Edwin Unger, "Group Theoretic Structures in the Fixed Charge Transportation Problem," presented to the national meeting of ORSA, New Orleans, 1972.

54. Walker, Warren E., "A Heuristic Adjacent Extreme Point Algorithm for the Fixed Charge Problem," Management Science 22,5 (1976), 587-596.

55. Walker, Warren, Michael Aquilina, Dennis Schur, "Development and Use of a Fixed Charge Programming Model for Regional Solid Waste Planning," ORSA/TIMS conference, San Juan, Puerto Rico, 1974.

56. Zangwill, Willard I., "Minimum Concave Cost Flows in Certain Networks," Management Science 14,7 (1968), 429-450.

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Center for Cybernetic Studies The University of Texas at Austin	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

406
197

3. REPORT TITLE

A New Optimization Method for Large-Scale Fixed Charge Transportation Problems

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

5. AUTHOR(S) (First name, middle initial, last name)

Richard S. Barr
Fred W. Glover
Darwin D. Klingman

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
March 1979	39	56

8a. CONTRACT OR GRANT NO. N00014-75-C-0616, N00014-75-C-0569	9a. ORIGINATOR'S REPORT NUMBER(S) Center for Cybernetic Studies Research Report CCS 350
b. PROJECT NO. NR047-021 c. d.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT

This document has been approved for public release and sale; its distribution is unlimited

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY

13. ABSTRACT

This paper presents a branch-and-bound algorithm for solving fixed charge transportation problems where not all cells exist. The algorithm exploits the absence of full problem density in several ways, thus yielding a procedure which is especially applicable to solving real-world problems which are normally quite sparse. Additionally, streamlined new procedures for pruning the decision tree and calculating penalties are presented. We present computational experience with both a set of large test problems and a set of dense test problems from the literature. Comparisons with other codes are uniformly favorable to the new method, which runs more than twice as fast as the best alternative.

DD FORM 1 NOV 65 1473 (PAGE 1)

S/N 0101-807-6811

Security Classification

A-31408

Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Networks						
Linear Programming						
Generalized Networks						

DD FORM 1 NOV 65 1473 (BACK)

S/N 0102-014-6800

Security Classification

A-31409